CLAIMS

1. A method for distributing high data rate output data to a number of different ports, comprising the steps of:

providing an output module having a plurality of different output channels;

providing a demultiplexer in communication with said output module for receiving said output channels; and

providing an output clock for synchronizing said output module with said demultiplexer.

2. The method of Claim 1, further comprising the step of:

bonding together any predetermined number of channels for each of said ports independently of each of said other ports.

3. The method of Claim 1, further comprising the step of:

providing a synchronization scheme in which a synchronization string is always written to a particular channel before said output channels are allowed to be clocked.

4. The method of Claim 3, further comprising the step of:

providing each output channel with its own output buffer;

wherein, once synchronization is established, each time said clock sends out a signal, a new word is put into an appropriate output buffer.

5. The method of Claim 4, further comprising the step of:

providing an address counter for controlling each said output buffer;

wherein said address counter waits for said synchronization string before counting through each of said output buffers.

6. The method of Claim 5, further comprising the step of:

providing a shift register associated with each said output buffer for receiving data in parallel as an input and for outputting said data in a serial fashion to an associated output buffer.

7. The method of Claim 6, further comprising the steps of:

connecting said shift registers together, such that by the time a first word is output from a first output buffer a next word may begin being output from said first output buffer; and

running a clock on said first output buffer and said next output buffer at a multiple of a rate that said clock would normally run for just said first output buffer.

8. A demultiplexer, comprising:

an output clock generator for transferring data from an output module to a demultiplexer input;

an address counter associated with said demultiplexer for targeting an output buffer associated with each of a plurality of output channels;

write logic associated with said demultiplexer for enabling current input data to be stored into an addressed output buffer;

a plurality of shift registers associated with said demultiplexer, one each connected to each output buffer for providing serial transmission of said stored data to an associated output.

9. The demultiplexer of Claim 8, wherein each of said shift registers has a carry output which is connected to a carry input of a shift register in a next lower numbered output channel.

10. The demultiplexer of Claim 9, further comprising:

clock generation logic for generating a plurality of different output clocks;

wherein each of said output clocks may be programmed to be an integral divisor of a reference clock;

wherein said reference clock may be either identical to an output clock provided to said output module, or it may be an integer multiple of said clock.

11. The demultiplexer of Claim 10, further comprising:

clock selection logic associated with each individual output buffer for enabling each output buffer to be run at one of a plurality of different clock rates, independently of said other output buffers.

12. The demultiplexer of Claim 8, wherein a channel is reserved for detection of synchronization information to ensure that data transferred between said output module and said demultiplexer is written to a proper output buffer.

13. The demultiplexer of Claim 12, wherein said address counter is inhibited from advancing past a first address until said reserved channel is written with a synchronization string.

14. A synchronization method for a demultiplexer, comprising the steps of:

initially creating a data stream with an output module to reset an address counter in said demultiplexer regardless of said demultiplexer's initial condition;

said output module writing a stream of multiplexed channel information with one word per channel, wherein a reserved channel is always written with a string that is used to initiate synchronization;

said output module interleaving data from all current output channels together, with one word per channel, wherein said output module always writes said string prior to issuing said data; and

providing an address counter that continues to hold at said reserved channel if at any time said reserved channel is written with any value other than said string, wherein a count is permitted to advance to an active output channel and data are output only after said address counter is written with said string.

15. The synchronization method of Claim 14, wherein data are repeatedly written to said reserved channel by said output module until said string appears in said data stream in the event that a loss of synchronization occurs.

16. A method for multiplexing a different number of channels onto each of a plurality of serial output streams, comprising:

providing a demultiplexer for enabling a number of channels that are bonded together to be set independently for each output channel;

said demultiplexer:

at system initialization, configuring each output channel to run at an appropriate clock rate;

selecting a same rate for sequential output channels; and

using a lowest numbered channel's output for actual data output, while ignoring remaining outputs.

17. An apparatus for distributing high data rate output data to a number of different ports in a system having an output module having a plurality of different output channels, said apparatus comprising:

a demultiplexer in communication with said output module for receiving said output channels; and

an output clock for synchronizing said output module with said demultiplexer.

18. The apparatus of Claim 17, further comprising:

means for bonding together any predetermined number of channels for each of said ports independently of each of said other ports.

19. The apparatus of Claim 17, further comprising:

a synchronization scheme in which a synchronization string is always written to a particular channel before said output channels are allowed to be clocked.

20. The apparatus of Claim 19, further comprising:

an output buffer each output channel;

wherein, once synchronization is established, each time said clock sends out a signal, a new word is put into an appropriate output buffer.

21. The apparatus of Claim 20, further comprising:

an address counter for controlling each said output buffer;

wherein said address counter waits for said synchronization string before counting through each of said output buffers.

22. The apparatus of Claim 21, further comprising:

a shift register associated with each said output buffer for receiving data in parallel as an input and for outputting said data in a serial fashion to an associated output buffer.

23. The apparatus of Claim 22, wherein said shift registers are connected together, such that by the time a first word is output from a first output a next word may begin being output from said first output; and further comprising:

a clock for clocking said first output and said next output, said clock running at a multiple of a rate that said clock would normally run for just said first output.

24. A method for demultiplexing, comprising the steps of:

transferring data from an output module to a demultiplexer input with an output clock generator;

targeting an output buffer associated with each of a plurality of output channels with an address counter associated with said demultiplexer;

enabling current input data to be stored into an addressed output buffer with write logic associated with said demultiplexer;

providing serial transmission of said stored data to an associated output with a plurality of shift registers associated with said demultiplexer, one each connected to each output buffer.

25. The method of Claim 24, wherein each of said shift registers has a carry output which is connected to a carry input of a shift register in a next lower numbered output channel.

26. The method of Claim 25, further comprising the step of:

generating a plurality of different output clocks with clock generation logic;

wherein each of said output clocks may be programmed to be an integral divisor of a reference clock;

wherein said reference clock may be either identical to an output clock provided to said output module, or it may be an integer multiple of said clock.

27. The method of Claim 26, further comprising the step of:

enabling each output buffer to be run at one of a plurality of different clock rates, independently of said other output buffers with clock selection logic associated with each individual output buffer.

28. The method of Claim 24, wherein a channel is reserved for detection of synchronization information to ensure that data transferred between said output module and said demultiplexer is written to a proper output buffer.

29. The method of Claim 28, wherein said address counter is inhibited from advancing past a first address until said reserved channel is written with a synchronization string.

30. A synchronization apparatus for a demultiplexer, comprising:

an output module for initially creating a data stream to reset an address counter in said demultiplexer regardless of said demultiplexer's initial condition;

said output module writing a stream of multiplexed channel information with one word per channel, wherein a reserved channel is always written with a string that is used to initiate synchronization;

said output module interleaving data from all current output channels together, with one word per channel, wherein said output module always writes said string prior to issuing said data; and

an address counter that continues to hold at said reserved channel if at any time said reserved channel is written with any value other than said string, wherein a count is permitted to advance to an active output channel and data are output only after said address counter is written with said string.

31. The synchronization apparatus of Claim 30, wherein data are repeatedly written to said reserved channel by said output module until said string appears in said data stream in the event that a loss of synchronization occurs.

32. An apparatus for multiplexing a different number of channels onto each of a plurality of serial output streams, comprising:

a demultiplexer for enabling a number of channels that are bonded together to be set independently for each output channel;

said demultiplexer comprising:

means for configuring each output channel to run at an appropriate clock rate at system initialization;

means for selecting a same rate for sequential output channels; and

means for using a lowest numbered channel's output for actual data output, while ignoring remaining outputs.